Amendments to the Specification:

Rewrite the paragraph at page 3, lines 16 to 24 as follows:

This invention describes an advanced high performance bus bridge, also known as AHB-to-HTB (High performance data Transfer Bus) bus bridge. The AHB-to-HTB bus bridge of this invention provides a means for the interfacing of two separate AHB-style busses allowing communication between them and securing data integrity. Since these busses have different characteristics, one for CPU support and the other for support of a large amount of data transfer by a single peripheral, the bus bridge is defined with clear master-slave protocol.

Rewrite the paragraph at page 6, line 16 to page 7, line 13 as follows:

Figure 2 illustrates the signal flow between a master requesting control of the AHB bus, the arbiter performing the arbitration decision and the slave selected by the master for a command to be executed in this standard AMBA system. AHB ~~bust~~ bus arbiter ~~111~~ 110, AHB master 200 and AHB slave 210 each receive a reset signal HResetx 222 and a clock signal HClockx 223. The AHB master 200 makes the request of AHB arbiter 110 by activating HBusReqx signal 231. The AHB master 200 receives permission from AHB arbiter 110 by HGrantx signal 232. The AHB master 200 confirms the grant and locks this arbitration decision by HLock signal 233. AHB master 200 then sends address 205 to AHB decoder 111. AHB decoder 111 activates a select signal 112 supplied to the selected slave device. In this example the selected slave device is AHB slave 210. The interaction of AHB master 200 and AHB slave 210 is completed via

the control signals 213 and acknowledged via HResp signal 211 and HReady signal 212. Data for read and write operations flows between all masters and all slaves via the AHB bus 100. AHB slave 210 supplies data to AHB bus 100 via HRData[31:0] bus 206 and receives data from AHB bus 100 via HWData[31:0] bus 207. Likewise, AHB master 200 receives data from AHB bus 100 via HRData[31:0] bus 208 and supplies data to AHB bus 100 via HWData[31:0] bus 209. Note in this regard that reads and writes are considered from the point of view of AHB master 200. Thus in a data read data flows from AHB slave 210 to AHB bus 100 via HRData[31:0] bus 206 and from AHB bus 100 via HRData[31:0] bus 208. Of course only one master is activated at a given time and this master selects only one slave on which it will execute a transfer (read or write) command.

Rewrite the paragraph at page 8, line 24 to page 9, line 4 as follows:

Next, AHB-to-HTB bus bridge 315 must win arbitration on the HTB bus 330. This is also shown as a multiplexing operation where multiplexer 420 under control of HTB arbiter/decoder ~~420~~ 421 selectively couples either AHB-to-HTB bus bridge 315 or HTB bus master 333 to HTB bus 330. During this period when the arbitrations are pending, AHB-to-HTB bus bridge 315 must hold AHB bus 300 while waiting for HTB arbitration. This can seriously degrade system performance since no activity will be occurring on AHB bus 300 during this period. AHB bus 300 will be the most active bus in most systems. To relieve this stall condition during a write condition, a write buffer is provided within AHB-to-HTB bus bridge 315.

Rewrite the paragraph at page 10, lines 8 to 22 as follows:

In order to write to a HTB bus peripheral, CPU 301 or DMA 303 must first be granted control of AHB bus 300 by AHB bus arbiter 410. Then AHB-to-HTB bus bridge 315 must be granted control of HTB bus 330 by HTB bus arbiter/decoder ~~420~~ 421. When the AHB-to-HTB bus bridge 315 is granted control of HTB bus 330, AHB-to-HTB bus bridge 315 will supply the address latched in address FIFO 510 to HTB bus arbiter/decoder 421. HTB bus arbiter/decoder 421 will decode this address to supply the necessary chip select signals analogous to select signal 112 illustrated in Figures 1 and 2. Since the entire system contains only one memory map, this will not cause any conflicts to other devices on other busses. When generating this address on HTB bus 330, AHB-to-HTB bus bridge 315 will follow standard AHB bus timings, pipelining the address one cycle before outputting the data.

Rewrite the paragraph at page 10, line 29 to page 11, line 14 as follows:

Referring again to Figure 5, when the first word is written to AHB-to-HTB bus bridge 315 from AHB bus 300, the full address will be latched into memory bus address latch 509 and data will be latched in memory bus data latch 519. When latched, the AHB-to-HTB bus bridge 315 will make a request HBusReqWrite 551 to the HTB Bus 330. A grant is acknowledged by grant signal HGrantx 553. If granted, the address in memory bus address latch 509 will be supplied to HAddr bus 511 and data in memory bus data latch 519 will be supplied to HData bus 521. This

supply may be via write buffers FIFOs 510 and 520 if these FIFOs contain data. Arbiter interface 505 will also generate HLockx signal 546 to HTB arbiter/decoder ~~420~~ 421. If not granted, the AHB-to-HTB bus bridge 315 can store more address and data in FIFOs 510 and 520 until these FIFOs are full. When the FIFOs 510 and 520 are full, AHB-to-HTB bus bridge 315 signals a not READY event 532 to the master on AHB bus 300.